REMARKS

Applicants respectfully traverse and request reconsideration.

Claims 2, 5 and 7 have been canceled while claims 1, 3, 6, 8-9, 12, 17, 21 and 23-24 have been amended. Claims 3, 6, 8, 9 and 24 have been amended to reflect a change in dependency. Claims 1 and 21 have been amended to incorporate the limitations present in canceled claims 2, 5 and 7. Claims 1 and 21 presently include, *inter alia*:

> a control block . . . wherein the control block receives information based on the operation code and generates control information provided to the plurality of accumulation registers and the selection block, wherein the control information . . . causes the selected accumulation register to store the result corresponding to the selected thread . . . ;
>
> a second operation unit . . . operably coupled to receive a third operand and a fourth operand . . . [and] produce a second operation result . . . ; and
>
> an arbitration module . . . [that] receives operation codes from a plurality of thread controllers corresponding to the plurality of threads . . . [and] determines order of execution of the operation codes received based on an application specific prioritization scheme.

Claims 12 and 23 have been amended to incorporate the limitations present in canceled claim 7. As amended, the method further includes "using an arbitration module to receive operation codes from a plurality of thread controllers corresponding to the plurality of threads and to determine the order of execution of the operation codes received based on an application specific prioritization scheme." Lastly, claim 17 has been amended such that the arbitration module determines the order of execution of the operation codes received "based on an application specific prioritization scheme."

Claims 1-10, 12-19, and 21-22 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,446,193 ("Alidina") in view of Gregory T. Byrd and Mark A. Holliday's "Multithreaded Processor Architectures" (Byrd). With respect to amended claim 1, Applicants respectfully note that no combination of Alidina or Byrd teaches or suggests, among other things, "an arbitration module operably coupled to the control block and the second operation unit, wherein the arbitration module receives operation codes from a plurality of thread controllers corresponding to the plurality of threads, and wherein the arbitration module determines order of execution of the operation codes received based on an application specific prioritization scheme." While the Office action references page 39 of Byrd as teaching an arbitration module, the cited reference appears to teach a plurality of hardware contexts and a context selection block different from Applicants' claimed invention. As described throughout the Byrd reference, a context is the machine state associated with a particular thread. (P. 40, "Defining Terms"). One context may represent a running or active thread, while other contexts may represent threads that are eligible to run or threads that are waiting on an operation to run. (Byrd, p. 38, caption for first figure). The context selection block performs context switching - the act of redirecting processor execution from one context to another, usually involving halting one thread's execution and starting the execution of another. (P. 40, ¶ 2; "Defining Terms"). In one instance, Byrd teaches that "switching between loaded threads should just require changing a context pointer and allowing any instructions in progress to complete." (Byrd, p. 42, ¶ 2)

The context selection block of Byrd does not appear to order the execution of operation codes based on an application specific prioritization scheme as claimed but rather teaches the generic ability of the context selection block to redirect an execution of one thread to the execution of another thread based on latency delays. (*See e.g.*, p. 40, ¶ 4). In other words, there does not appear to be an organized or ordered queue of threads based on an application specific prioritization scheme. Because Byrd appears to teach a system where the context selection block can merely allow any instruction in progress to complete without regard to a predetermined order, the reference does not teach or suggest Applicants' claimed subject matter. For this reason alone, claim 1 is believed to be allowable. If the Examiner maintains this rejection, Applicants respectfully request a showing in Byrd where the context selection block determines the order of the execution of operation codes based on an application specific prioritization scheme as claimed.

Additionally, Applicants respectfully note that no combination of Alidina or Byrd teach or suggest "a plurality of accumulation registers . . . wherein each accumulation register of the plurality of accumulation registers corresponds to one of the plurality of threads, [and] wherein a selected accumulation register stores the first operation result . . . . As admitted by the Office action, Alidina does not teach multi-threading and having registers and operands which correspond to each individual thread. (Office action, p. 3, ¶ 6). However, the Office action appears to characterize Byrd as teaching the incorporation of registers and operands in a multi-threaded environment where each register corresponds to each individual thread. (Byrd, pp. 38-40). Applicants respectfully submit that Byrd does not appear to teach using the general-purpose registers (which partially comprise a hardware context) to store a first operation result as claimed. Instead, Byrd teaches that general-purpose registers can be used to hold the machine state of a particular thread. (Byrd, p. 40, "Defining Terms"). Because the general-purpose registers of Byrd are only used to store the machine state of a thread and does not appear to store an accumulation or other operation result as designed in the Byrd system, the general-purpose registers fail to teach or suggest Applicants' claimed accumulation registers.

With respect to Applicants' claim language directed at "a selection block operably coupled to a plurality of accumulation registers . . . wherein the selection block selects the second operand provided to the first operation unit from a set of potential operands, [and] wherein the set of potential operands includes contents of each accumulation register of the plurality of accumulation registers," Applicants respectfully note that the Office action appears to reference a split multiplexer (SMUX) of Alidina that selects inputs of the accumulation registers (Fig. 3, element 30). As illustrated in Alidina, Figure 3, the SMUX receives inputs from various saturators associated with an arithmetic logic unit, an add unit and a bit manipulation unit. The results as modified from these various units are fed through a split multiplexer to a register array comprising eight 40 bit accumulators. (Alidina, column 4, lines 63-66). As a consequence, the SMUX appears to select inputs for the accumulation registers and does not appear to teach or suggest a selection block that, among other things, selects a second operand for a first operation unit wherein the set of potential operands includes contents of each accumulation register.

With respect to Applicants' claim limitation directed at "a control block . . . [which] receives information based on operation code and generates control information . . . wherein the control information . . . causes the selected accumulation register to store the result," the Office action cites Alidina, column 5, lines 8-18. However, Applicants respectfully note that this portion of Alidina appears to teach an inapposite process wherein preselected mode bits are provided by control registers to selectively provide feedback along a path to multiplier registers. (Emphasis added). In other words, at least this cited portion of the Alidina reference makes clear that the preselected mode bits do not cause the selected accumulation registers to store a result. Therefore, if the Examiner maintains this rejection, Applicants respectfully request a showing where the control bits in Byrd: (1) are generated based on operation code and (2) cause the selected accumulation register to store the result.

Additionally, Applicants respectfully repeat the relevant remarks made in the previous response filed March 21, 2005 with respect to the improper motivation to combine the teachings of Alidina and Byrd. For instance, Alidina repeatedly describes performing concurrent operations on a parallel processing system rather than the execution of operations on a single processor in a multi-threaded environment. (*See e.g.*, col. 1, ll. 13-14; col. 2, ll. 41-42; col. 3, ll. 39-73). As described in Alidina, the use of a parallel processing system reduces the number of instruction cycles and enhances the number of instructions the system can process per second. (Col. 3, ll. 49-52). In contrast, Byrd teaches the use of multiple threads in a system having only one processor such that the system can fill its otherwise idle cycles by switching threads and thereby boosting performance. (*See e.g.*, p. 28-40).

Thus, while both Alidina and Byrd sought ways to improve processing performance, the two systems are mutually exclusive because they address different aspects of increasing performance. For instance, Alidina's system addresses reducing instruction cycles by increasing the number of instructions executed per cycle through the use of a second processor. Byrd, however, is not concerned with cycles themselves but rather is only concerned with decreasing idle time (e.g., due to latency) on a single processor. Each overcomes their respective problem using unique architectures without suggesting the need to incorporate components from the other's system. Thus, one of ordinary skill in the art would not be motivated to combine select components of a system using a parallel architecture with select components of a non-parallel, multi-threaded system without using Applicants' specification as a roadmap. For these reasons in addition to those articulate above, Applicants respectfully submit claim 1 for allowance.

With respect to claim 12, Applicants respectfully repeat the relevant remarks made above with respect to claim 1. Additionally, Applicants highlight the new limitation added to claim 12 wherein the method further includes "using an arbitration module to receive operation codes from a plurality of thread controllers corresponding to the plurality of threads and to determine the order of execution of the operation codes received based on an application specific prioritization scheme." Applicants further note that claim 12 contains, among other novel and nonobvious subject matter, limitations directed at selecting a selected accumulation registers from a plurality of accumulation registers and storing the first result in the selected accumulation register. Because no combination of Alidina or Byrd teaches or suggests, among other novel and nonobvious subject matter, the use of an arbitration module or the selection and storing of a first result in a selected accumulation register as claimed by Applicants, Applicants respectfully submit claim 12 for allowance.

With respect to claim 17, Applicants repeat the relevant remarks made above with respect to claim 1 noting once again that no combination of Alidina and Byrd teaches or suggests an arbitration module, a plurality of accumulation registers and a selection block as claimed and described above. For this reason, Applicants respectfully present claim 17 for allowance.

With respect to dependent claim 8, the Office action appears to reference Alidina, column 2, lines 57-60 as teaching a multi-thread accumulation circuit including a vector engine that performs at least one of dot product operations, vector multiply accumulate operations, vector addition operations and vector multiplication operations. However, the cited reference of Alidina, merely describes and teaches the use of a vectored register in a digital signal processor of a type that may include a plurality of accumulators and not a vector engine. (Emphasis added). As described in Alidina, the accumulators in such a system may include respective high and low parts where the vectored register has a first part from a first of the accumulators and a second part from a second of the accumulators. The result is that the first and second parts cooperate to define a vector subjected to processing as if it were a single register. (Alidina, col. 2, ll. 57-65). For instance, Applicants respectfully draw the Examiner's attention to Alidina, Figure 2 wherein a vector register is illustrated by reference numeral 12. (Col. 3, l. 66 – col. 4, l. 2). By utilizing vector registers, Alidina teaches that each of the two high parts can be processed concurrently thereby reducing in half the number of instruction cycles otherwise required to process the parts. (Alidina, col. 4, l. 11-16). While this motivation and description adequately describe a vectored register, the teaching fails to teach or suggest a vector engine as claimed. Because no combination of Alidina or

Byrd anticipates or renders obvious the claimed limitations, Applicants respectfully believe that claim 8 is in proper condition for allowance.

With respect to claim 9, the Office action appears to reference various portions of Alidina that allegedly teach or suggest a memory operably coupled to the selection block, the first operation unit, and the collection block, wherein the memory stores the first operation result produced by the first operation unit, and wherein contents of the memory are selectively included in the set of potential operands based on a portion of the control information generated by the control block. While Alidina, Figure 3 and the cited references specified in the Office action appear to describe a variety of product registers and accumulation registers, Applicants respectfully request a showing where Alidina specifically teaches or suggests the use of a memory operably coupled to the selection block, the first operation unit and the control block and wherein the contents of the memory are selectively included in the set of potential operands (of the first operation unit) as claimed by Applicants. Because no combination of Alidina or Byrd appears to teach or suggest the memory as claimed, Applicants respectfully submit claim 9 for allowance.

With respect to dependent Claims 3, 4, 6, 10, 13-16, 18-19 and 22, Applicants respectfully repeat the relevant remarks made above with respect to independent claims 1, 12, 17 and 21. Applicants further note that the above-listed dependent claims contain novel, non-obvious and patentable subject matter and are in proper condition for allowance.

Dependent claims 11 and 20 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Alidina in view of Byrd and further in view of U.S. Patent No. 5,673,377 ("Berkaloff"). Applicants respectfully repeat the relevant remarks made above with respect to independent claims 1 and 17 and further note that claims 11 and 20 add additional novel, non-obvious and patentable subject matter. Additionally, Applicants respectfully note that Berkaloff describes a system and method for providing a three-dimensional graphic-user interface operator real-time feedback to modification to three-dimensional objects projected onto a computer display screen. (*See* Abstract). The reference appears to describe that various shading parameters such as diffuse color and specular color allow for the parameterization of the perceived color of an object. Putting aside the fact that no combination of Alidina and Byrd teaches Applicants' claimed invention, the mere discussion of these shading parameters in a system for providing feedback in real time to an operator where the system is insufficient to render Applicants' claimed subject matter obvious to one of ordinary skill in the art. In fact, the cited reference of Berkaloff fails to teach or suggest an architectural structure as employed by the system. For these reasons and for those listed in the previous response filed March 21, 2005, Applicants respectfully submit claims 11 and 20 for allowance.

Claim 23 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Alidina in view of U.S. Patent No. 5,933,627 ("Parady"). Applicants respectfully repeat the relevant remarks made above with respect to independent claims 1, 12, 17 and 21. Applicants further note that Parady appears to teach thread switching after cache misses using registers in an identical manner as described in Byrd. As described, Parady fails to teach or suggest the use of an arbitration module as described above in the context of the claimed invention. For this reason and because claim 23 further contains novel, non-obvious and patentable subject matter, Applicants respectfully request its allowance.

With respect to dependent claim 24, Applicants respectfully repeat the relevant remarks made above with respect to independent claim 23. Applicants further note that claim 24 contains novel, non-obvious and patentable subject matter and is in proper condition for allowance.

Applicants respectfully submit that the claims are in condition for allowance and that a timely Notice of Allowance be issued in this case. The Examiner is invited to contact the below-listed attorney if the Examiner believes that a telephone conference will advance the prosecution of this application.

Respectfully submitted,

Dated: 10/17/05

By: C Reckamp
Christopher J. Reckamp
Reg. No. 34,414

Vedder, Price, Kaufman & Kammholz, P.C.
222 North LaSalle Street
Chicago, Illinois 60601
Telephone: (312) 609-7599
Facsimile: (312) 609-5005